

## Mouton William

L'application permet de gérer la base de données d'un restaurant avec des tables gérant les clients ou encore le stock.

La base de données étant créée avec les différentes tables, il fallait maintenant réussir à relier cette base à une interface. Le sql étant natif à python cela s'est avéré moins difficile que prévu. Or, il a fallu intégrer toutes les idées que j'avais pour gérer les bases de données ou encore différents « events ».

## Schéma relationnel :

7 tables :

-stock

(id\_plat, nom\_plat, quantite\_plat)

-réservation

( id\_client\*, id\_table\*, date, heure\_debut, heure\_fin)

-fidélité

(id\_client\*, réduction)

-facture

(id\_facture, id\_client\*, total)

-client

(id\_client, nom, prenom, argent\_depense)

-tables

(id\_table, numero\_table)

-menu

(id\_menu, id\_plat\*, prix\_plat)

gras = clé primaire

\* = clé étrangère

\_\_\_\_\_ = contrainte

4 views :

-Afficher le plat qui a été le plus stocké.

-Table qui a le plus de place.

-Client ayant le plus dépensé.

- Plat le plus cher.

4 déclencheurs :

- Après avoir ajouté une réservation, les quantités de chaque plat dans le stock est augmenté de 100.

- Après avoir supprimé un client, sa réservation est également supprimée.

- Après avoir supprimé une réservation, le client est également supprimé.

- Après avoir modifié un client, son id\_table est augmenté de 1.